

UUU	UUU	EEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEE	TTT	PPP

RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	44	44
RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	44	44
RR RR	RR	MMMM	MMMM	SS	TT	EE	SS	TT	44
RR RR	RR	MMMM	MMMM	SS	TT	EE	SS	TT	44
RR RR	RR	MM MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM MM	MM	SS	TT	EE	SS	TT	44
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	4444444444	
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	4444444444	
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	44
RR RR	RR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	44

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

```
0000 1      .IDENT 'V04-000'  
0000 78     $BEGIN RMSTEST4,009, __RMSTEST,<XAB RMS TEST PROGRAM>,<GBL, LONG>  
0000 79     ;  
0000 80     ;  
0000 81     ;  
0000 82     .ENABL DBG  
0000 83     ;  
0000 84     ;  
0000 85     ; macros:  
0000 86     ;  
0000 87     ;  
0000 88     ;  
0000 89     ;  
0000 90     ;  
0000 91     .MACRO TYPE STRING, ?L  
0000 92     STORE <STRING>  
0000 93     BLBC VERTBOSITY,L  
0000 94     MOVL #$.TMPX, CMDORAB+RABSL RBF  
0000 95     MOVW #$.TMPX1, CMDORAB+RABSW RSZ  
0000 96     $PUT RAB=CMDORAB,ERR=REPORT_ERROR  
0000 97     BSBW ERR  
0000 98     L: .ENDM TYPE  
0000 99     ;  
0000 100    ;  
0000 101    ;  
0000 102    ;  
0000 103    .MACRO STORE STRING,PRE  
0000 104    .SAVE  
0000 105    .PSECT --$RMSNAM  
0000 106    $.TMPX=.  
0000 107    PRE ; store any carriage control info  
0000 108    .ASCII %STRING%  
0000 109    $.TMPX1=-$.TMPX  
0000 110    .RESTORE  
0000 111    .ENDM STORE
```

0000	113	
0000	114	:
0000	115	
0000	116	.MACRO BEGIN TSTNAM
0000	117	STORE <TSTNAM>
0000	118	MOVL #SS.TMPX,BEG_DESCR+4 : addr
0000	119	MOVL #SS.TMPX1,BEG_DESCR : len
0000	120	BSBW BEGPUT
0000	121	.ENDM BEGIN
0000	122	.MACRO FINISH TSTNAM
0000	123	STORE <TSTNAM>
0000	124	MOVL #SS.TMPX,FIN_DESCR+4 : addr
0000	125	MOVL #SS.TMPX1,FIN_DESCR : len
0000	126	BSBW FINPUT
0000	127	.ENDM FINISH
0000	128	.MACRO FIELD FLDNAM
0000	129	STORE <FLDNAM>
0000	130	MOVL #SS.TMPX,FLD_DESCR+4 : addr
0000	131	MOVL #SS.TMPX1,FLD_DESCR : len
0000	132	BSBW FLDPUT
0000	133	.ENDM FIELD

```

00000000 135 .PSECT RMSTEST, GBL, LONG
0000 136 .ALIGN C0NG
0000 137 T4START:::
0000 138 T4FAB:: $FAB FNM=<TST$DISK:T4FILE.DAT;1>,-
0000 139 ORG=SEQ,-
0000 140 RFM=VFC,-
0000 141 RAT=CR,-
0000 142 FSZ=4,-
0000 143 MRS=100,-
0000 144 NAM=NAMBLK,-
0000 145 DEQ=12
00,0 146 FLUSH_FAB:::
0050 147 $FAB FAC=<PUT, GET>,-
0050 148 FNM=<TST$DISK:T4FILE.DAT;1>,-
0050 149 NAM=NAMBLK,-
0050 150 SHR=<PUT, GET, UPI>,-
0050 151 XAB=FHCXAB
00A0 152
00A0 153 :
00A0 154 : attention: in order to assemble this module, t4fab and FLUSH_RAB
00A0 155 : have been put into another module, RMSTESTR
00A0 156 :
00A0 157 :
00A0 158 FHCXAB:::
00A0 159 $XABFHC NXT=ALQXAB
00CC 160 ALQXAB:::
00CC 161 $XABALL NXT=PROXAB,-
00CC 162 DEQ=15
00EC 163 PROXAB:::
00EC 164 $XABPRO
0144 165 DATXAB:::
0144 166 $XABDAT
0170 167 RDTXAB:::
0170 168 $XABRDT
0184 169 TRMXAB:::
0184 170 $XABTRM
01A8 171 $RMSDEFEND
00000024 01A8 172 EXTRA=XABSL_SBN-4 ; 4 bytes of extra (spare) char.
01A8 173 SAVEPRO: .WORD 0 ; word to save pro in
0000 01A8 174
01AA 175
01AA 176 :
01AA 177 : THESE ARE THE DATA STRUCTURES FOR DATE AND TIME XAB CHECKS
01AA 178 :
01AA 179 :
20 33 36 39 31 2D 52 41 4D 2D 33 20 01AA 180 CDT: .ASCII / 3-MAR-1963 03:03:03.03/
33 30 2E 33 30 3A 33 30 3A 33 30 01B6 181 CDTL=-CDT
20 34 39 31 2D 52 50 41 2D 34 20 01C1 182 RDT: .ASCII / 4-APR-1944 04:04:04.04/
34 30 2E 34 30 3A 34 30 3A 34 30 01CD 183 RDTL=-RDT
20 38 39 31 2D 47 55 41 2D 38 20 01D8 184 EDT: .ASCII / 8-AUG-1988 08:08:08.08/
38 30 2E 38 30 3A 38 30 3A 38 30 01E4 185 EDTL=-EDT
20 38 34 39 31 2D 43 45 44 2D 32 31 01EF 186 RDT2: .ASCII / 12-DEC-1948 12:12:12.12/
32 31 2E 32 31 3A 32 31 3A 32 31 01FB 187 RDTL2=-RDT2
00000017 0206

```

0000001AA'0C000017	0206	188	CDTDEC: .LONG	CDTL,CDT	
000001C1'00C00017	020E	189	RDTDEC: .LONG	RDTL,RDT	
000001D8'00000017	0216	190	EDTDEC: .LONG	EDTL,EDT	
000001EF'00000017	021E	191	RDT2DEC:		
	0226	192	.LONG	RDTL2,RDT2	
0000	0226	193			
0000	0226	194	CURRVN: .WORD	0	
0000	0228	195	LEN: .WORD	0	
00000243	022A	196	CMPDAT: .BLKB	25	; length of returned string
	0243	197	CMPDATDEC:		; has room for longest possible date
0000022A'00000019	0243	198	.LONG	25,CMPDAT	
00000000	0248	199	CURRDT: .LONG	0	; address of current rdt string
00000000	024F	200	UIC: .LONG	0	; room to save current uic
00000008	0253	201	DATLEN=11		; length of date
00000014	0253	202	TIMLEN=20		; length of ascii date and time
	0253	203			
	0253	204			

0FFC 0253 206 RMTTEST_4A::
 0253 207 .WORD ^MCR2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
 0255 208 BEGIN <XAB TESTS>
 026A 209
 026A 210
 026A 211 : create a file -- sys\$disk:t4file.dat:1 -- with controlled attributes
 026A 212 : and write 1 record to it, to further control the attributes
 026A 213 :
 026A 214
 5B FD92 CF DE 026A 215 MOVAL T4FAB,R11 ; r11 will be fab throughout
 026F 216 ; initialize values for restartability
 026F 217 SFAB_STORE FAB=R11,-
 026F 218 SHR=<PUT,GET,UP>,-
 026F 219 FOP=<SUP,CTG>,-
 026F 220 XAB=FHCXAB,-
 026F 221 ALQ=#0,-
 026F 222 FAC=PUT
 0289 223
 FE1D CF B4 0289 224 CLRW XABSW_LRL+FHCXAB ; make sure lrl is 0
 028D 225 \$XABPRO_STORE XAB=PROXAB,-
 028D 226 PRO=<RWED,RWED,RD,RWED>,-
 028D 227 UIC=<333,44>
 AB AF 0C A0 D0 02A2 228 MOVL XABSL_UIC(R0),UIC ; save it for checking
 FEFB CF 08 A0 B0 02A7 229 MOVW XABSW_PRO(R0),SAVEPRO ; ditto
 02AD 230 \$XABALL_STORE XAB=ALQXAB,-
 02AD 231 ALQ=#10,-
 02AD 232 AOP=CTG
 02BB 233 \$CREATE FAB=R11,- ; with all xabs linked in
 02BB 234 ERR=REPORT_ERROR
 FD33' 30 02CA 235 BSBW ERR
 00000000'8F E1 02CD 236 BBC #DEV\$V SQD,-
 03 40 AB 02D3 237 FABSL_DEV(R11),10\$
 03F7 31 02D6 238 BRW MTA
 OA FDFF CF D1 02D9 239 10\$: CMPL ALQXAB+XABSL_ALQ,#10 ; if mta, skip this
 15 15 18 02DE 240 BGEQ RIGHT ; allocated 10 blks?
 02E0 241 FIELD <ALQ IN XAB (NOT = DESIRED ALLOC ON CREATE)>
 02F5 242 RIGHT:
 15 04 AB 15 E1 02F5 243 BBC #FABSV_CBT,FABSL_FOP(R11),OK1
 15 04 AB 14 E0 02FA 244 FIELD <CBT BIT SET, THEREFORE>
 030F 245 OK1: BBS #FABSV_CTG,FABSL_FOP(R11),OK2
 0314 246 FIELD <CTG BIT CLEAR, THEREFORE>
 0329 247 OK2:
 04FB 30 0329 248 BSBW ZERO_XABS
 032C 249 \$DISPLAY FAB=R11,-
 FCC2' 30 032C 250 ERR=REPORT_ERROR
 0501 30 033B 251 BSBW ERR
 033E 252 BSBW CHECK CR ; check xabs
 0341 253 TYPE <OK AFTER CREATE AND DISPLAY>
 0370 254
 0370 255 :
 0370 256 :extend
 037U 257 :
 0370 258 :
 FD67 CF 30 D0 0370 259 MOVL #48,XABSL_ALQ+ALQXAB ; set up
 FD57 CF D4 0375 260 CLRL XABSL_NXT+ALQXAB
 24 AB FD4F CF DE 0379 261 MOVAL ALQXAB,FABSL_XAB(R11)
 037F 262 \$EXTEND FAB=R11,- ; extend file after create

		037F	263	ERR=REPORT_ERROR	PS
		038E	264		--
		038E	265		.
		038E	266	:using alq from xab	SA
		038E	267	:	SR
		038E	268		--
30	FD47	FC6F'	30	038E 269 BSBW ERR	Ph
	CF		D1	0391 270 CMPL ALQXAB+XAB\$L_ALQ,#48 ; alq in xab should ret actual alq	--
	15		18	0396 271 BGEQ ALQOK	In
				0398 272 FIELD <ALQ IN XAB (NOT = DESIRED ALLOC ON EXTEND)>	Co
				03AD 273 ALQOK:	Pa
				03AD 274	Sy
				03AD 275 :	Pa
				03AD 276 :undo damage to xab links	Sy
				03AD 277 :	Ps
				03AD 278	Cr
24	AB	FCEF	CF	DE 03AD 279 MOVAL FHCXAB,FAB\$L_XAB(R11)	As
FCEA	CF	FD15	CF	DE 03B3 280 MOVAL ALQXAB,XAB\$L_NXT+FHCXAB	Th
FDOF	CF	FD2E	CF	DE 03BA 281 MOVAL PROXAB,XAB\$L_NXT+ALQXAB	80
				03C1 282 \$CONNECT RAB=T4RAB,-	Th
				03C1 283 ERR=REPORT_ERROR	67
32	41	8F	FC29'	30 03D4 284 BSBW ERR	Th
		6E	00	2C 03D7 285 MOVCS #0,(SP),#^A/A/,#50,CPYBUF; move 50 a's into cpybuf	67
		00000000'EF		03DD	
00000000'EF		00000000'EF		DE 03E2 286 MOVAL CPYBUF,RAB\$L_RBF+T4RAB	10
00000000'EF		32	B0	03ED 287 MOVW #50,RAB\$W_RSZ+T4RAB	Th
				03F4 288 \$PUT RAB=T4RAB,-	MA
				03F4 289 ERR=REPORT_ERROR	
				FBF6' 30 0407 290 BSBW ERR	

			040A	292				
			040A	293				
			040A	294	: take some time to try out flush			
			040A	295	:			
			040A	296				
			040A	297	\$FLUSH RAB=T4RAB,-			
			040A	298	ERR=REPORT_ERROR			
		FBE0'	30	041D	299	BSBW	ERR	
		0404	30	0420	300	BSBW	ZERO_XABS	
				0423	301	\$OPEN	FAB=FLUSH_FAB,-	
				0423	302		ERR=REPORT_ERROR	
		FBC9'	30	0434	303	BSBW	ERR	
		0418	30	0437	304	BSBW	CHECK_XABS	
				043A	305	\$CONNECT	RAB=FLUSH_RAB,-	
				043A	306		ERR=REPORT_ERROR	
		FBBO'	30	044D	307	BSBW	ERR	
				0450	308	\$GET	RAB=FLUSH_RAB,-	
				0450	309		ERR=REPORT_ERROR	
				FB9A'	310	BSBW	ERR	
		32	00000000'EF	B1	0466	CMPW	RABSW_RSZ+FLUSH_RAB,#50 ; got right rec. size	
				15	13	BEQL	RSZ OR	
					046D	FIELD	<RSZ IN RAB>	
					046F			
					0484	314	RSZ_OK:	
					0484	315	CMPC5 #50,CPYBUF,#^A/A/,#0,(SP); is record ok?	
					6E	048E		
					15	13	BEQL	
					048F	316	YES	
					0491	317	FIELD <RECORD>	
					04A6	318		
					04A6	319	\$GET FLUSH_RAB	
					04A6	320	CMPL R0,#R\$\$_EOF	: this should be eof
					04A6	321	BEQL 10\$	
					04B3	322	MOVAL FLUSH_RAB,R10	
					04BA	323	BSBW EOFPUT	
					04BC	324	10\$:	
					04C3	325	CLRL FABSL_XAB+FLUSH_FAB	: no xabs on close for now
					04C3	326	\$DISCONNECT RAB=FLUSH_RAB,-	: clean up after flush
					04C6	327	ERR=REPORT_ERROR	
					FBAA CF	04C6		
					04CA	328	BSBW	ERR
					04CA	329	\$CLOSE	FAB=FLUSH_FAB,-
					04CA	330		ERR=REPORT_ERROR
					FB20'	30	04DD	: all done w/ flush test
					04EO	331	BSBW	ERR
					04EO	332	TYPE <ALL DONE WITH FLUSH TEST>	
					0523	333		
					0523	334		
					0523	335	: all done with flush test	
					0523	336		
					0523	337		

```

      0523 339
      0523 340
      0523 341 $DISCONNECT RAB=T4RAB,-
      0523 342 ERR=REPORT_ERROR
      0523 343 BSBW ERR
      0523 344 $CLOSE FAB=R11,-
      0523 345 ERR=REPORT_ERROR
      0523 346 BSBW ERR
      0523 347
      0523 348 CLRB FAB$B SHR(R11)
      0523 349 MOVAL FHCXAB,FAB$L_XAB(R11) ; set up xab links again
      0523 350 BSBW ZERO_XABS
      0523 351 $OPEN FAB=R11,-
      0523 352 ERR=REPORT_ERROR
      0523 353 BSBW ERR
      0523 354
      0523 355 BBC #FAB$V_CBT,FAB$L_FOP(R11),CC
      0523 356 FIELD <CBT BIT WAS SET, THEREFORE>
      0523 357 CC: BBC #FAB$V_CTG,FAB$L_FOP(R11),OK; after extend, not ctg
      0523 358 FIELD <CTG BIT WAS SET, THEREFORE>
      0523 359
      0523 360 :
      0523 361 : check fhc xab
      0523 362 :
      0523 363
      02B2 30 059D 364 OK: BSBW CHECK_XABS
      05A0 365
      05A0 366 :
      05A0 367 :do another extend, forcing it to get the value from the alq of the fab
      05A0 368 :
      05A0 369
      24 AB 04 05A0 370 CLRL FAB$L_XAB(R11)
      10 AB 0C 00 05A3 371 MOVL #12,FAB$L_ALQ(R11)
      05A7 372 $EXTEND FAB=R11,-
      05A7 373 ERR=REPORT_ERROR
      FA47' 30 05B6 374 BSBW ERR
      05B9 375 CMPL FAB$L_ALQ(R11),#12 ; alq in fab=12
      10 AB D1 05B9 376 BGEQ ALQOKT
      15 18 05BD 377 FIELD <ALQ IN FAB (NOT = DESIRED ALLOCATION AFTER EXTEND)>
      05D4 378 ALQOK1:
      05D4 379
      05D4 380 :
      05D4 381 :change protection and uic on close
      05D4 382 :
      05D4 383
      05D4 384 SXABPRO_STORE XAB=PROXAB,-
      05D4 385 PRO=<RWED,RWED,RED,RWED>,-
      05D4 386 UIC=<222,55>
      24 AB 50 00 05E9 387 MOVL R0,FAB$L_XAB(R11) : set up xab
      FBB5 CF 08 A0 B0 05ED 388 MOVW XABSW_PRO(R0),SAVEPRO : for check
      FC56 CF 0C A0 D0 05F3 389 MOVL XAB$L_UIC(R0),UIC : ditto
      05F9 390 $CLOSE FAB=RT1,-
      05F9 391 ERR=REPORT_ERROR
      F9F5' 30 0608 392 BSBW ERR
      060B 393 MOVAL FHCXAB,FAB$L_XAB(R11)
      24 AB FA91 CF DE 060B 394 $OPEN FAB=R11,-
      0611 395 MOVAL FAB$L_XAB(R11) ; check changes after ext
      0611 396 ERR=REPORT_ERROR

```

15 04 AB F9DD' 30 0620 396 BSBW ERR
 15 04 AB 15 E1 0623 397 BBC #FABSV_CBT,FABSL_FOP(R11),NOCBT
 15 04 AB 14 E1 0628 398 FIELD <CBT BIT WAS SET, THEREFORE>
 15 04 AB 14 E1 0630 399 NOCBT: BBC #FABSV_CTG,FABSL_FOP(R11),NOCTG ; shouldn't be ctg, after extend
 15 04 AB 14 E1 0642 400 FIELD <CTG BIT WAS SET, AFTER 2 EXTENDS, THEREFORE>
 00000046 59 FA45 CF DE 0657 401 NOCTG: MOVAL FHCXAB,R9 : check pertinent fields
 8F 0C A9 D1 065C 402 CMP_XABSL_HBK(R9),#70 ; alq=10+48+12
 15 18 0664 403 BGEQ HBKOK
 00 28 A9 D1 0666 404 FIELD <HBK IN FHCXAB (AFTER 2ND EXTEND)>
 15 13 067B 405 HBKOK: CMPL XABSL_SBN(R9),#0 ; not ctg anymore
 00 28 A9 D1 067F 406 BEQL STILL_OK
 15 13 0681 407 FIELD <SBN IN FHCXAB (AFTER 2ND EXTEND)>
 01C9 30 0696 408 STILL_OK:
 00 46 0699 409 BSBW CHECK_ALL
 025E 30 069B 410 .BYTE 70,0 ; values for alq,ctg (not ctg)
 025E 30 069E 411 BSBW CHECK_PRO
 24 AB D4 06CD 412 TYPE <DONE WITH 2ND EXTEND, NOW TEST DATES>
 06D0 413 CLRL FABSL_XAB(R11) ; no xabs on this close, for now
 06D0 414 MTA: \$CLOSE FAB=RT1,- ; continue if mta
 F91E' 30 06DF 415 BSBW ERR=REPORT_ERROR
 06E2 416 BSBW ERR
 06E2 417 BSBW ERR
 06E2 418 :
 06E2 419 :before finishing up, have some fun with the dat and rdt xabs
 06E2 420 :
 06E2 421 :
 06E2 422 :
 59 FASE CF DE 06E2 423 MOVAL DATXAB,R9
 06E7 424 SBINTIM_S CDTDEC,XABSL_CDT(R9)
 06F5 425 SBINTIM_S RDTDEC,XABSL_RDT(R9)
 0703 426 SBINTIM_S EDTDEC,XABSL_EDT(R9)
 08 A9 00C8 8F B0 0711 427 MOVW #200,XABSL_RVN(R9)
 FB08 CF 00C8 8F B0 0717 428 MOVW #200,CURRVN
 24 AB 59 D0 071E 429 MOVL R9,FABSL_XAB(R11)
 04 A9 D4 0722 430 CLRL XABSL_NXT(R9)
 FB1F CF FA98 CF DE 0725 431 MOVAL RDT,CURRDT
 04 AB 00000080 8F C8 072C 432 BISL #FABSM_RWO,FABSL_FOP(R11) ; current rdt str
 0734 433 \$CREATE FAB=R1T,- ; rewind if mta
 0734 434 BSBW ERR=REPORT_ERROR
 F9FB CF FA26 CF 30 0743 435 BSBW ERR
 00C6 30 0746 436 MOVAL RDTXAB,XABSL_NXT+DATXAB
 0750 437 BSBW ZERO_DAT_XABS
 0750 438 \$DISPLAY FAB=R11,-
 0750 439 BSBW ERR=REPORT_ERROR
 F89E' 30 075F 440 BSBW ERR
 032B 30 0762 441 BSBW CHECK_DATES
 FADF CF FA86 CF DE 0765 442 MOVAL RDT2,CURRDT ; get a new rdt
 59 FA00 CF DE 076C 443 MOVAL RDTXAB,R9
 08 A9 012C 8F B0 0771 444 SBINTIM_S RDT2DEC,XABSL_RDT(R9)
 FA9A CF 012C 8F B0 077F 445 MOVW #300,XABSL_RVN(R9)
 24 AB 59 D0 0785 446 MOVW #300,CURRVN
 0790 447 MOVL R9,FABSL_XAB(R11) ; only rdt for close
 0790 448 \$CLOSE FAB=R11,-
 0790 449 BSBW ERR=REPORT_ERROR
 24 AB F85E' 30 079F 450 BSBW ERR
 F99E CF DE 07A2 451 MOVAL DATXAB,FABSL_XAB(R11) ; only dat for open
 F99C CF D4 07A8 452 CLRL XABSL_NXT+DATXAB

68 10 07AC 453 BSB8 ZERO_DAT XABS
07AE 454 \$FAB_STORE FAB=R11,- ; can't "put" to mta
07AE 455
07B2 456
07B2 457
24 AB F83C' 30 07C1 458 SOPEN FAB=R11,-
F9AB (F DE 07C4 459
07CA 460 BSBW ERR
07CA 461 MOVAL RDTXAB,FAB\$L_XAB(R11) ; get rdt on display
F824' 30 07D9 462 \$DISPLAY FAB=R11,-
02B1 30 07DC 463 BSBW ERR
24 AB D4 07DF 464 CLRL FAB\$L_XAB(R11) ; no xabs for this close
07E2 465 \$CLOSE FAB=RT1,-
07E2 466 BSBW ERR
F80C' 30 07F1 467
07F4 468
07F4 469
07F4 470 \$FAB_STORE FAB=R11,- ; restore fac
07F4 471 BICL #FABSM_RWO,FAB\$L_FOP(R11) ; and fop
04 AB 00000080 BF CA 07F8 472 FINISH <XAB TESTS>
0800 473
04 0815 474 RET

F92C CF	26	00	6E	00	2C	0816	476	
F950 CF	0E	00	6E	00	2C	0816	477	
					05	0816	478	;2 routines to zero out the xabs before checking the results
						0816	479	:
						0816	480	
						0816	481	ZERO_DAT_XABS:
						0816	482	
						0816	483	:
						0816	484	;zero out the dat and rdt xabs
						0816	485	:
						0816	486	
						0816	487	MOVCS #0,(SP),#0,#<XAB\$C_DATLEN-6>,DATXAB+6
						081E	488	MOVCS #0,(SP),#0,#<XAB\$C_RDTLEN-6>,RDTXAB+6
						0826	489	RSB
						0827	490	
						0827	491	ZERO_XABS:
						0827	492	
						0827	493	:
						0827	494	;zero out the fhc, all and pro xabs
						0827	495	:
						0827	496	
F877 CF	26	00	6E	00	2C	0827	497	MOVCS #0,(SP),#0,#<XAB\$C_FHCLEN-6>,FHCXAB+6
F89B CF	1A	00	6E	00	2C	082F	498	MOVCS #0,(SP),#0,#<XAB\$C_ALLLEN-6>,ALQXAB+6
0052 8F	00	6E	00	2C	0837	499	MOVCS #0,(SP),#0,#<XAB\$C_PROLEN-6>,PROXAB+6	
F8B1 CF					05	083E		
						0841	500	RSB

```

0842 502 CHECK_CR:
0842 503
0842 504 :
0842 505 ;routine to check xabs after create and subsequent displ
0842 506 :
0842 507
00F7 30 0842 508 BSBW CHECK_FHC
U1 00 0A 00 0845 509 .BYTE 0,10,0,1 ; values for lrl,alq,ffb,sbn
0016 30 0849 510 BSBW CHECK_ALL
01 0A 084C 511 .BYTE 10,1 ; values for alq,ctg ( should be)
00AB 30 084E 512 BSBW CHECK_PRO
05 0851 513 RSB
0852 514
0852 515 CHECK_XABS:
0852 516
0852 517 :
0852 518 ;general routine to check out all xabs
0852 519 :
0852 520
00E7 30 0852 521 BSBW CHECK_FHC
00 38 3A 32 0855 522 .BYTE 50,58,56,0 ; values for lrl,alq,ffb,sbn(not ctg)
0006 30 0859 523 BSBW CHECK_ALL
00 3A 085C 524 .BYTE 58,0 ; values for alq,ctg ( not ctg anymore)
009B 30 085E 525 BSBW CHECK_PRO
05 0861 526 RSB
0862 527
0862 528 CHECK_ALL:
0862 529
0862 530 :
0862 531 ;routine to check out the allocation xab
0862 532 :
0862 533
59 F866 CF DE 0862 534 MOVAL ALQXAB,R9
OF 14 A9 B1 0867 535 CMPW XAB$W_DEQ(R9),#15
15 13 086B 536 BEQL DEQOK
00 16 A9 91 0882 537 DEQOK: FIELD <DEQ IN ALL. XAB>
15 13 0886 538 CMPB XAB$B_BKZ(R9),#0
0888 539 BEQL BKZOK
50 00 BE 9A 089D 540 FIELD <BKZ IN ALL. XAB>
6E D6 08A1 541 BKZOK: MOVZBL @(SP),R0
10 A9 50 D1 08A3 542 INCL (SP)
15 15 08A7 543 CMPL R0,XAB$L_ALQ(R9)
08A9 544 BLEQ ALQOK2
50 00 BE 9A 08BE 545 ALQOK2: FIELD <ALQ IN ALL. XAB>
6E D6 08C2 546 MOVZBL @(SP),R0
1A 50 E9 08C4 547 INCL (SP)
2F 08 A9 07 E0 08C7 548 BLBC R0,NOTCTG
08C0 549 BBS #XAB$V CTG,XAB$B_AOP(R9),AOPOK ; should be set
15 08 A9 07 E1 08E1 550 FIELD <CTG C|R IN AOP, THEREFORE>
08E6 551 NOTCTG: BBC #XAB$V CTG,XAB$B_AOP(R9),AOPOK ; should be clear
05 08FB 552 FIELD <CTG SET IN AOP, THEREFORE>
08FC 553 AOPOK: RSB
08FC 554
08FC 555 CHECK_PRO:
08FC 556
08FC 557 :
08FC 558 ;check the protection xab

```

			08FC	559 :				
			08FC	560				
08	A9	F7EC	CF	DE	08FC	561	MOVAL	PROXAB,R9
		F8A3	CF	15	B1	0901	CMPW	SAVEPR0,XAB\$W_PRO(R9) ; cmp to saved value
					13	0907	BEQL	PROCK
0C	A9	F92D	CF	15	D1	091E	FIELD	<PROT FIELD IN PROT XAB>
					13	0924	CMPL	UIC,XAB\$L_UIC(R9)
						0926	BEQL	UICOK
						093B	FIELD	<UIC FIELD IN PROT. XAB>
						093C	RSB	

```

093C 571 CHECK_FHC:
093C 572
093C 573 :
093C 574 ; check fhc xab carefully
093C 575 :
093C 576

59 F760 CF DE 093C 577 MOVAL FHCXAB,R9 ; r9 is ptr to xab thru-out cmp's
03 08 A9 91 0941 578 EXTC: CMPB XAB$B_RFO(R9),#FAB$C_VFC; check rec. format & org.
15 13 0945 580 BEQL RFOC
09 A9 02 93 095C 581 FIELD <RFO IN FHC XAB>
15 12 0960 582 RFOC: BITB #FAB$M_CR,XAB$B_ATR(R9) ; check rat field
0962 583 BNEQ ATRC
50 00 BE 9A 0977 584 FIELD <ATR IN FHC XAB>
6E D6 0978 585 ATRC: MOVZBL @(SP),R0
0A A9 B1 097D 586 INCL (SP)
15 13 0981 587 CMPW XAB$W_LRL(R9),R0 ; check longest record len
0983 588 BEQL LRLC
50 00 BE 9A 0998 590 LRLC: MOVZBL @(SP),R0
6E D6 099C 591 INCL (SP)
50 0C A9 D1 099E 592 CMPL XAB$L_HBK(R9),R0 ; check alg
15 18 09A2 593 BGEQ HBKC
09A4 594 FIELD <HBKC IN FHC XAB>
01 10 A9 D1 09B9 595 HBKC: CMPL XAB$L_EBK(R9),#1 ; check end block
15 13 09BD 596 BEQL EBKC
09BF 597 FIELD <EBKC IN FHC XAB>
50 00 BE 9A 09D4 598 EBKC: MOVZBL @(SP),R0
6E D6 09D8 599 INCL (SP)
50 14 A9 B1 09DA 600 CMPW XAB$W_FFB(R9),R0 ; check first free byte
15 13 09DE 601 BEQL FFBC ; its len of rec + fsz + 2
09E0 602 FIELD <FFB IN FHC XAB>
00 16 A9 91 09F5 603 FFBC: CMPB XAB$B_BKZ(R9),#0 ; check bucket size
15 13 09F9 604 BEQL BKZC
09FB 605 FIELD <BKZ IN FHC XAB>
04 17 A9 91 0A10 606 BKZC: CMPB XAB$B_HSZ(R9),#4 ; check fixed area size
15 13 0A14 607 BEQL HSZC
0A16 608 FIELD <HSZ IN FHC XAB>
0064 8F 18 A9 B1 0A2B 609 HSZC: CMPW XAB$W_MRZ(R9),#100 ; check max. rec size
15 13 0A31 610 BEQL MRZC
0A33 611 FIELD <MRZ IN FHC XAB>
0F 1A A9 B1 0A48 612 MRZC: CMPW XAB$W_DXQ(R9),#15 ; check def ext. qty
15 13 0A4C 613 BEQL DXQC
0A4E 614 FIELD <DXQ IN FHC XAB>
50 00 BE 9A 0A63 615 DXQC: MOVZBL @(SP),R0
6E D6 0A67 616 INCL (SP)
08 50 E9 0A69 617 BLBC R0,10$ ; make sure non-zero lbn
00 28 A9 D1 0A6C 618 CMPL XAB$L_SBN(R9),#0
0A70 619
0A70 620 ; since it's ctg
0A70 621 :
0A70 622 :
0A70 623 :
1D 12 0A70 624 BNEQ FHC_OK
06 11 0A72 625 BRB 20$ ; make sure zero lbn
00 28 A9 D1 0A74 626 10$: CMPL XAB$L_SBN(R9),#0 ; since it isn't contig.
15 13 0A78 627 BEQL FHC_OR

```

RMSTEST4
009

XAB RMS TEST PROGRAM

D 10

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 15
(13)

05 0A7A 628 20\$: FIELD <SBN IN FHC XAB>
05 0A8F 629 FHC_OK: RSB

```

0A90 631
0A90 632 CHECK_DATES:
0A90 633
0A90 634 :
0A90 635 ;routine to check edt and cdt in dat xab
0A90 636 ;and rdt and rvn in both dat and rdt xab's
0A90 637 :
0A90 638

05 40 AB F793 CF 0B 80 0A90 639      MOVW #DATLEN,LEN      : default is check date only
00000000'8F E0 0A95 640      BBS #DEV$V_SQD,FABSL_DEV(R11),10$ ; if not mta, check date and time
F785 CF 14 80 0A9E 641      MOVW #TMLER,LEN
59 F69D CF DE 0AA3 642 10$: MOVAL DATXAB,R9
F766 CF F6E9 CF F76A CF 29 0ABA 643      $ASCTIM_S ,CMPDATDEC,XAB$Q_CDT(R9)
15 13 0AC4 644      CMPC3 LEN,CDT,CMPDAT
0AC6 645      BEQL CDTOK
0ADB 646      FIELD <CDT IN DAT XAB>
0ADB 647 CDTOK:
F735 CF F6E6 CF 0B 29 0AED 648      $ASCTIM_S ,CMPDATDEC,XAB$Q_EDT(R9)
15 13 0AF5 649      CMPC3 #DATLEN,EDT,CMPDAT ; only check date
0AF7 650      BEQL EDTOK
0B0C 651      FIELD <EDT IN DAT XAB>
01 40 AB 00000000'8F E1 0B0C 652 EDTOK:
05 0B15 653      BBC #DEV$V_SQD,FABSL_DEV(R11),10$ ; that's it if mta
F6F8 CF F71C DF F6FC CF 29 0B16 654 10$: $ASCTIM_S ,CMPDATDEC,XAB$Q_RDT(R9)
15 13 0B28 655      CMPC3 LEN,ACURRDT,CMPDAT
0B32 656      BEQL RDTOK
0B34 657      FIELD <RDT IN DAT XAB>
0B49 659 RDTOK:
F6C4 CF F6E8 DF F6C8 CF 29 0B5C 660      $ASCTIM_S ,CMPDATDEC,XAB$Q_RDT+RDTXAB
15 13 0B66 661      CMPC3 LEN,ACURRDT,CMPDAT
0B68 662      BEQL RDTOK1
0B70 663      FIELD <RDT IN RDT XAB>
08 A9 F6A5 CF B1 0B7D 664 RDTOK1:
15 13 0B83 665      CMPW CURRVN,XAB$W_RVN(R9)
0B85 666      BEQL RVNOK
F5D7 CF F688 CF B1 0B9A 667 RVNOK: FIELD <RVN IN DAT XAB>
15 13 0BA1 668      CMPW CURRVN,XAB$W_RVN+RDTXAB
0BA3 669      BEQL RVNOK1
05 0BB8 670      FIELD <RVN IN RDT XAB>
0BB9 671 RVNOK1: RSB
0BB9 672 .END

```

\$\$PSECT_EP	= 00000000		ERR	***** X 01
\$\$TAB	= 00000184	R D 01	EXTC	00000941 R D 01
\$\$TABEND	= 000001A8	R D 01	EXTRA	= 00000024 D
\$\$TMP	= 00000001		FAB\$B_FAC	= 00000016 D
\$\$TMP1	= 00000002		FAB\$B_FNS	= 00000034 D
\$\$TMP2	= 0000005B		FAB\$B_SHR	= 00000017 D
\$\$TMP5	= 00000002		FAB\$C_BID	= 00000003 D
\$\$TMPX	= 00000352	R D 04	FAB\$C_BLN	= 00000050 D
\$\$TMPX1	= 0000000E		FAB\$C_SEQ	= 00000000 D
\$\$RMSTEST	= 0000001E		FAB\$C_VAR	= 00000002 D
\$\$RMS_PBUGCHK	= 00000010		FAB\$C_VFC	= 00000003 D
\$\$RMS_TBUGCHK	= 00000008		FAB\$L_ALQ	= 00000010 D
\$\$RMS_UMODE	= 00000004		FAB\$L_DEV	= 00000040 D
..AFLG	= 00000000	D	FAB\$L_FNA	= 00000020 D
..FLG	= 00000002	D	FAB\$L_FOP	= 00000004 D
..MOD	= 00000001	D	FAB\$L_XAB	= 00000024 D
..N	= 00000001		FAB\$M_CR	= 00000002 D
..TYP	= 00000003	D	FAB\$M_RWO	= 00000080 D
.LEN	= 00000001		FAB\$V_CBT	= 00000015 D
ALQOK	000003AD	R D 01	FAB\$V_CHAN_MODE	= 00000002 D
ALQOK1	000005D4	R D 01	FAB\$V_CR	= 00000001 D
ALQOK2	000008BE	R D 01	FAB\$V_CTG	= 00000014 D
ALQXAB	000000CC	RG D 01	FAB\$V_FILE_MODE	= 00000004 D
AOPOK	000008FB	RR D 01	FAB\$V_GET	= 00000001 D
ATRC	00000977	R D 01	FAB\$V_LNM_MODE	= 00000000 D
BEGPUT	***** X 01		FAB\$V_PUT	= 00000000 D
BEG_DESCR	***** X 01		FAB\$V_SUP	= 00000002 D
BKZC	00000A10	R D 01	FAB\$VUPI	= 00000006 D
BKZOK	0000089D	R D 01	FAB\$W_GBC	= 0000048 D
CC	00000583	R D 01	FFBC	000009F5 R D 01
CDT	000001AA	R D 01	FHC_XAB	000000A0 RG D 01
CDTDEC	00000206	R D 01	FHC_OK	00000A8F R D 01
CDTL	= 00000017		FINPUT	***** X 01
CDTOK	00000ADB	R D 01	FIN_DESCR	***** X 01
CHECK_ALL	00000862	RR D 01	FLDPUT	***** X 01
CHECK_CR	00000842	RR D 01	FLD_DESCR	***** X 01
CHECK_DATES	00000A90	RR D 01	FLUSH_FAB	00000050 RG D 01
CHECK_FHC	0000093C	RR D 01	FLUSH_RAB	***** X 01
CHECK_PRO	000008FC	RR D 01	HBKC	000009B9 R D 01
CHECK_XABS	00000852	R D 01	HBKOK	0000067B R D 01
CMDORAB	***** X 01		HSZC	00000A2B R D 01
CMPDAT	0000022A	R D 01	LEN	00000228 R D 01
CMPDATDEC	00000243	R D 01	LRLC	00000998 R D 01
CPYBUF	***** X 01		MRZC	00000A48 R D 01
CURRDT	0000024B	R D 01	MTA	000006D0 R D 01
CURRVN	00000226	R D 01	NAMBLK	***** X 01
DATLEN	= 00000008		NOCBT	0000063D R D 01
DATXAB	00000144	RG D 01	NOCTG	00000657 R D 01
DEQOK	00000882	R D 01	NOTCTG	000008E1 R D 01
DEV\$V_SQD	***** X 01		OK	0000059D R D 01
DXQC	00000A63	R D 01	OK1	0000030F R D 01
EBKC	000009D4	R D 01	OK2	00000329 R D 01
EDT	000001D8	R D 01	PROOK	0000091E R D 01
EDTDEC	00000216	R D 01	PROXAB	000000EC RG D 01
EDTL	= 00000017		RAB\$L_RBF	***** X 01
EDTOK	0000080C	R D 01	RAB\$W_RSZ	***** X 01
EOFPUT	***** X 01		RDT	000001C1 R D 01

RDT2
 RDT2DEC
 RDTDEC
 RDTL
 RDTL2
 RDTOK
 RD1OK1
 RDTXAB
 REPORT_ERROR
 RFOC
 RIGHT
 RMSS_EOF
 RMSTEST_4A
 RSZ_OK
 RVNOK
 RVNOK1
 SAVEPRO
 STILL_OK
 SYSSA5CTIM
 SYSSBINTIM
 SYSSCLOSE
 SYSSCONNECT
 SYSSCREATE
 SYSSDISCONNECT
 SYSSDISPLAY
 SYSEXEND
 SYSSFLUSH
 SYSSGET
 SYSSOPEN
 SYSSPUT
 T4FAB
 T4RAB
 T4START
 TIMLEN
 TRMXAB
 UIC
 UICOK
 VERBOSITY
 XAB\$B_AID
 XAB\$B_AOP
 XAB\$B_ATR
 XAB\$B_BKZ
 XAB\$B_HSZ
 XAB\$B_MTACC
 XAB\$B_PROT_MODE
 XAB\$B_PROT_OPT
 XAB\$B_RFO
 XAB\$C_ALL
 XAB\$C_ALLLEN
 XAB\$C_DAT
 XAB\$C_DATLEN
 XAB\$C_FHC
 XAB\$C_FHCLEN
 XAB\$C_PRO
 XAB\$C_PROLEN
 XAB\$C_RDT
 XAB\$C_RDTLEN

= 000001EF R D 01	XAB\$C_TRM	= 0000001F D	\$1
= 0000021E R D 01	XAB\$C_TRMLEN	= 00000024 D	\$1
= 0000020E R D 01	XAB\$L_ACLBUF	= 00000018 D	\$1
= 00000017 R D 01	XAB\$L_ACLCTX	= 00000020 D	\$1
= 00000017 R D 01	XAB\$L_ALQ	= 00000010 D	\$1
= 00000849 R D 01	XAB\$L_EBK	= 00000010 D	\$1
= 0000087D R D 01	XAB\$L_HBK	= 0000000C D	\$1
= 00000170 RG D 01	XAB\$L_ITMLST	= 00000008 D	\$1
***** X 01	XAB\$L_LOC	= 0000000C D	\$1
= 0000095C R D 01	XAB\$L_NXT	= 00000004 D	\$1
= 000002F5 R D 01	XAB\$L_SBN	= 00000028 D	\$1
***** X 01	XAB\$L_UIC	= 0000000C D	\$1
= 00000253 RG D 01	XAB\$Q_CDT	= 00000014 D	BE
= 00000484 R D 01	XAB\$Q_EDT	= 0000001C D	BE
= 00000B9A R D 01	XAB\$Q_RDT	= 0000000C D	BK
= 00000B88 R D 01	XAB\$V_CTG	= 00000007 D	EF
= 000001A8 R D 01	XAB\$W_ACLSIZ	= 0000001C D	FI
= 00000696 R D 01	XAB\$W_DEQ	= 00000014 D	FI
***** GX 01	XAB\$W_DXQ	= 0000001A D	FI
***** GX 01	XAB\$W_FFB	= 00000014 D	FI
***** GX 01	XAB\$W_GRP	= 0000000E D	FI
***** GX 01	XAB\$W_ITMLST_LEN	= 0000000C D	FI
***** GX 01	XAB\$W_LRL	= 0000000A D	FI
***** GX 01	XAB\$W_MBM	= 0000000C D	FI
***** GX 01	XAB\$W_MRZ	= 00000018 D	FI
***** GX 01	XAB\$W_PRO	= 00000018 D	FI
***** GX 01	XAB\$W_RF10	= 00000018 D	FI
***** GX 01	XAB\$W_RF12	= 0000001A D	FI
***** GX 01	XAB\$W_RF14	= 0000001C D	FI
***** GX 01	XAB\$W_RVN	= 00000008 D	FI
= 00000000 RG D 01	XAB\$W_VOL	= 0000000A D	FA
***** X 01	YES	000004A6 R D 01	FA
= 00000000 RG D 01	ZERO_DAT_XABS	00000816 R D 01	FA
= 00000014 RG D 01	ZERO_XABS	00000827 R D 01	FA
= 00000184 RG D 01			
0000024F R D 01			
0000093B R D 01			
***** X 01			
= 00000017 D			
= 00000008 D			
= 00000009 D			
= 00000016 D			
= 00000017 D			
= 0000000A D			
= 00000010 D			
= 00000008 D			
= 00000008 D			
= 00000014 D			
= 00000020 D			
= 00000012 D			
= 0000002C D			
= 0000001D D			
= 0000002C D			
= 00000013 D			
= 00000058 D			
= 0000001E D			
= 00000014 D			

```
! Psect synopsis !
```

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMTEST	00000BB9 (3001.)	01 (1.)	NOPIC USR CON REL GBL NOSHR EXE RD WRT NOVEC LONG
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$RMSNAM	0000002A (42.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
--\$RMSNAM	00000360 (864.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

```
! Performance indicators !
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.43
Command processing	105	00:00:00.63	00:00:02.75
Pass 1	362	00:00:17.15	00:00:38.17
Symbol table sort	0	00:00:00.53	00:00:01.16
Pass 2	148	00:00:03.68	00:00:08.25
Symbol table output	24	00:00:00.15	00:00:00.31
Psect synopsis output	3	00:00:00.02	00:00:00.30
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	673	00:00:22.25	00:00:51.37

The working set limit was 1350 pages.

80832 bytes (158 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 460 non-local and 9 local symbols.

672 source lines were read in Pass 1, producing 53 object records in Pass 2.

67 pages of virtual memory were used to define 50 macros.

```
! Macro library statistics !
```

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB:1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB:2	50
TOTALS (all libraries)	50

1077 GETS were required to define 50 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSTEST4/OBJ=OBJ\$:RMSTEST4 MSRC\$:RMSTEST4/UPDATE=(ENH\$:RMSTEST4)+EXECMLS/LIB

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

